

Análisis y Generación de Cargas de Trabajo para la Evaluación de Estrategias de Calendarización en un Grid Computacional

¹Elisa Salazar, ²Andrei Tchernykh

¹Universidad de Sonora, Hermosillo, Sonora, México,
elizaliz@correo.uson.mx

²Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, BC México,
chernykh@cicese.mx

Resumen— El desempeño en el manejo de recursos en un Grid Computacional (C-Grid) depende de varios factores entre estos la carga de trabajo. Usualmente, son usados dos recursos de carga de trabajo para la evaluación de las estrategias de calendarización para computadoras paralelas: logs recolectados de sistemas reales y modelos de carga de trabajo. Sin embargo las soluciones de calendarización de trabajos para una sola computadora paralela y para un cluster difieren significativamente de las soluciones de calendarización en un Grid. Las aplicaciones de C-Grid funcionan en un amplio rango de dominio, cada una con sus propias características y requerimientos, las cuales representan a un amplio rango de usuarios. Por lo tanto las aplicaciones pueden ser diferentes entre máquinas específicas, sus cargas de trabajo y las posibles computadoras paralelas del C-Grid. Las cargas de trabajo que están basadas en logs recolectados de sistemas paralelos específicos no representan usuarios del C-Grid, y pueden afectar la exactitud de la simulación y la evaluación de las estrategias de calendarización para el ambiente de C-Grid. En este artículo, para superar estas dificultades proponemos un modelo de la carga para la simulación del C-Grid.

Palabras Clave—Grid, Logs, Modelos, Carga de trabajo, Calendarización, Manejo de recursos, Optimización.

I. INTRODUCCIÓN

El término Grid fue elegido como una analogía a las plantas eléctricas, donde estas proporcionan un servicio estable y dependiente a los usuarios sin importar su fuente, esto es similar al Grid porque puede proporcionar ciertos servicios como compartir una gran variedad de recursos computacionales para almacenamiento de grandes datos, o para la ejecución de programas que requieran un gran poder de cómputo.

En los últimos años, el cómputo paralelo se ha extendido como un instrumento de trabajo del Grid para las aplicaciones que demandan un poder de cómputo intensivo y se ha vuelto parte del llamado Grid Computacional (C-Grid). Con el C-Grid, los negocios pueden optimizar recursos de cómputo y de datos, combinar grandes cargas de trabajo que sobrepasen la capacidad de sus computadoras, compartir recursos a través de

redes de una manera transparente y confiable.

Para proporcionar estos servicios, se necesitan mecanismos para repartir el trabajo de forma automática y eficiente entre la gran variedad de recursos. La calendarización nos va a permitir que las colas de espera se reduzcan o crezcan, esto depende de los usuarios, las cargas de trabajo enviadas por estos y las estrategias empleadas para calendarizar.

Usualmente, son usados dos recursos de carga de trabajo para computadoras paralelas para la evaluación de las estrategias de calendarización: logs recolectados de sistemas reales y modelos de carga de trabajo. Sin embargo las soluciones de calendarización de trabajos para una sola computadora paralela y para un cluster pueden diferir significativamente de las soluciones de calendarización en un Grid.

El sistema de calendarización tiene que ser responsable de seleccionar las computadoras adecuadas para los trabajos de los usuarios tomando en consideración parámetros de trabajos, parámetros de cada computadora, requerimientos de la infraestructura de calendarización del C-Grid, y criterios de optimización.

II. GRID COMPUTACIONAL

2.1 Modelos de Grids

El modelo de arquitectura de Grid para simular sistemas manejadores de recursos [1] es influenciado por la manera en la que el calendarizador está estructurado. La estructura del calendarizador depende del número de recursos en los cuales los trabajos son calendarizados y el dominio en el cual los recursos son localizados. Hay tres modelos diferentes para estructuras de calendarizadores:

Modelo de calendarización centralizada: Este puede ser usado para manejar uno o múltiples recursos localizados en uno o múltiples dominios. Solamente puede dar soporte a políticas uniformes y es muy bueno para administrar sistemas de clusters (o colas por lotes) como el Condor, LSF y Condine. No es adecuado para sistemas manejadores de recursos de grid debido a que en ellos se espera políticas impuestas por los dueños de los recursos.

Modelo de calendarización descentralizada: En este modelo los calendarizadores interactúan entre ellos con el objetivo de decidir cual recurso debe ser aplicado a los

trabajos que estén en ejecución. En este esquema, no hay ningún líder central responsable de la calendarización, por tanto este modelo parece ser altamente escalable y tolerante a fallas. Este esquema es adecuado para sistemas de grid ya que los dueños de los recursos pueden definir sus políticas de calendarización. De cualquier manera, debido a que el estado de los trabajos y recursos remotos no están disponibles en una sola ubicación, la generación de calendarios óptima no es segura. Este modelo es difícil de implementar en un ambiente grid porque los dueños de los dominios de los recursos no están de acuerdo en una política global para el manejo de recursos.

Modelo de calendarización jerárquica: Este modelo es adecuado para sistemas de grid porque permite a los dueños de recursos remotos aplicar sus propias políticas sobre usuarios externos. Este modelo parece como si fuera un modelo híbrido (una combinación de los modelos centralizado y descentralizado), pero parece más un modelo centralizado y funciona bien para sistemas de grid. En este esquema el calendarizador que se encuentra más arriba en la jerarquía es llamado super-calendarizador o resoucer broker, el cual interactúa con los calendarizadores locales para de esta manera decidir los calendarios.

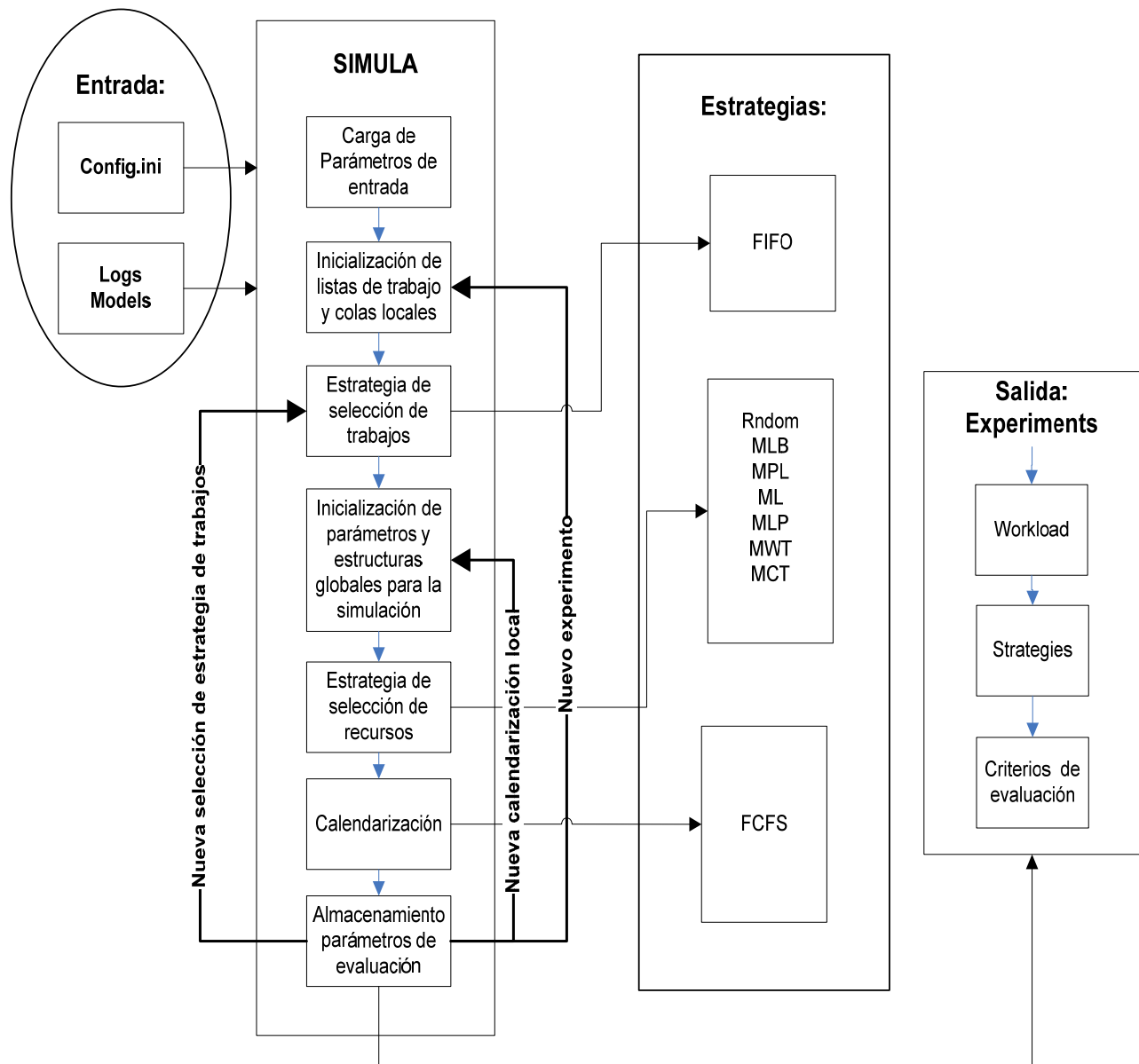


Figura 1. Estructura de Simula

Estos tres modelos son los más comunes y los más implementados en los sistemas de simulación de manejo de recursos.

2.2 Simulación de los Grids

Actualmente la simulación es la única manera posible para que los sistemas distribuidos de recursos heterogéneos con diferentes estrategias de selección de recursos puedan ser evaluados, además de que es una manera efectiva de evaluar sistemas cuando se trabaja con problemas de gran escala que involucran un gran número de recursos y usuarios. También hace posible explorar diferentes tipos de sistemas operando bajo una variedad de cargas de trabajos y algoritmos. La simulación ha sido usada extensamente para el modelado y evaluación de sistemas del mundo real, desde procesos de negocios y fábricas de líneas de ensamblaje hasta diseño de sistemas computacionales. A través del tiempo, la simulación se ha vuelto una importante disciplina y se han creado aplicaciones estándar y tecnologías de simulación.

Existen pocas herramientas disponibles para simulaciones de aplicaciones de calendarización en ambientes computacionales Grid. Los más conocidos son: Bricks [2], MicroGrid [3], SimGrid [4] y GridSim[5]. Actualmente en CICESE se está desarrollando un sistema de simulación de manejo de recursos en un ambiente grid "Simula". Simula cuenta con múltiples estrategias de selección de recursos para calendarización, las cuales están conformadas no solo por los típicos algoritmos de calendarización.

La Figura 1 muestra la estructura básica de Simula. Se puede observar que consiste de dos entradas básicas externas para trabajar (encerradas en un ovalo). La primera es un archivo de configuración llamado Config.ini, el cual tiene un formato establecido para la configuración de parámetros del simulador. La segunda entrada es un archivo de carga de trabajo, que es con la que trabajara el simulador para calendarizar, simulando la distribución de las tareas y el manejo de los recursos en el ambiente C-Grid.

El archivo de carga contiene información relevante para la simulación: una lista de tareas cada una descrita por ciertos parámetros como son el número de tarea, tiempo de inicio, tiempo de terminación, tiempo de ejecución, número de procesadores que ocupa esa tarea, etc. Cada conjunto de tareas contenido en un archivo es conocido como Log o Modelos, Los Logs son una recopilación de trabajos realizados en centros de investigación o de desarrollo reales como la NASA e IBM. Los modelos son programas para generar un conjunto de tareas, nace de un modelo de carga basado en análisis de logs reales. El recuadro más grande con título Simula representa como funciona internamente Simula, a la derecha de este recuadro están las estrategias y en el último recuadro se muestra la salida de los resultados.

2.3 Estrategias

Para la selección de trabajos, la estrategia a utilizar es

“primero en llegar, primero en salir” (FIFO por sus siglas en inglés First In, First Out).

La calendarización consiste de dos partes, la selección de la máquina con base en algún criterio y la calendarización local de tareas paralelizables. En la Tabla 1 se presentan siete estrategias de selección que utilizamos para la asignación de las tareas en los nodos del C-Grid.

Estrategia de selección de recursos	Descripción
MCT (Min Completion Time)	El menor tiempo de terminación posible es determinado basándose en un calendario parcial de trabajos ya asignados.
MLB (por sus siglas en inglés Min Lower Bound)	Selecciona un nodo con límite inferior del tiempo de ejecución de trabajos asignados anteriormente. Está basada únicamente en los parámetros de los trabajos ya asignados disponibles para el broker, y no necesita una construcción parcial de un calendario en cada nodo.
MLP (Min Load per-proc)	Selecciona el nodo con la mínima carga por procesador (número de trabajos sobre el número de procesadores en el nodo).
MPL (Min Parallel Load)	Selecciona un nodo con la carga promedio de cada procesador que sea mínima (suma de los tamaños de los trabajos sobre el número de procesadores en el nodo).
MWT (Min Waiting Time)	Selecciona un nodo con el mínimo promedio de tiempo de espera de trabajo.
Random	Selecciona un nodo arbitrariamente con distribución uniforme.

Tabla 1. Estrategias de selección de recursos.

En este artículo como estrategia de calendarización local utilizamos la estrategia FCFS (por sus siglas en inglés *First-Come-First-Serve*) donde el calendarizador inicia el trabajo en el orden en que se encuentra localmente. Si no hay suficientes recursos, el calendarizador espera la ejecución de las tareas ya asignadas hasta que se liberen los recursos necesarios.

2.4 Criterios de evaluación

Para la evaluación del desempeño de las diferentes estrategias usamos criterios comunes de evaluación, los cuales los dividimos en tres grupos: criterios del algoritmo de calendarización, del sistema y del usuario.

De acuerdo al criterio del algoritmo de calendarización:

Coficiente de desempeño (performance ratio). Es la medida de rendimiento, definida como $\rho = C_{max} / C_{opt}$, donde C_{max} es el tiempo de la finalización de todas las tareas, y $C_{opt} \geq \max(W_{tot} / m, p_{max})$ el tiempo mínimo posible para

terminarlas donde p_{\max} es el tiempo de ejecución máxima de una tarea, m el tamaño del nodo y W_{tot} es el trabajo total del C-Grid.

De acuerdo al criterio del sistema:

Utilización. Se define como la fracción del tiempo en la cual el C-Grid fue utilizado $U = W_{tot} / (W_{tot} + W_{idle})$, donde W_{tot} es el trabajo total y W_{idle} los recursos que no fueron utilizados durante la ejecución.

Rendimiento del procesamiento (throughput). Es el número de tareas finalizadas por unidad de tiempo n/C_{max} .

De acuerdo al criterio del usuario:

Tiempo de respuesta promedio (mean turnaround time). Es el promedio de finalización de todas las tareas desde que

entran hasta que salen: $\sum_{j=1}^n t_t^j / n$, donde $t_t^j = C^j - r^j$, C^j es

el tiempo de finalización de la tarea j , r^j es el tiempo de entrega de la tarea j . El criterio es importante desde el punto de vista del usuario porque su minimización se obtiene de minimizar el tiempo de respuesta promedio de la calendarización del grupo de trabajo.

Tiempo de espera (waiting time). Esta definido como el tiempo promedio de espera de las tareas antes de iniciar su

ejecución, entonces $\sum_{j=1}^n t_w^j / n$, donde $t_w^j = t_b^j - r^j$, t_b^j es

tiempo de inicio de ejecución de la tarea j .

Coefficiente de respuesta (response ratio). Está definido como el promedio de los coeficientes de respuesta de todas las

tareas $\frac{1}{n} \sum_{j=1}^n (t_w^j + p_s^j) / p_s^j$, donde el coeficiente de respuesta

para cada tarea es $(t_w^j + p_s^j) / p_s^j$, p_s^j es el tiempo de ejecución y t_w^j es el tiempo de espera de la tarea j .

III. CARGA DE TRABAJO

Usualmente, son usados dos recursos de carga de trabajo para la evaluación de las estrategias de calendarización: logs recolectados de sistemas paralelos de gran escala, como por ejemplo el NASA Ames iPSC 860, el SDSC Paragon, CTC SP2, etc., y modelos de carga de trabajo: Feitelson96, Lublin99, etc. Estos son representativos de varios archivos de carga de trabajo real y están basados en observaciones de centros de cómputo específicos.

Las aplicaciones de un C-Grid funcionan para una gran variedad de maquinas y usuarios, cada uno con sus propias características y requerimientos. Por lo tanto una aplicación puede tener diferencias entre máquinas específicas, sus cargas de trabajo y las posibles computadoras paralelas del C-Grid. Para simular el ambiente computacional usamos tres tipos de cargas de trabajo: logs, modelos, y carga sintética.

Los logs son registros de cargas recolectados de computadoras en uso, y tenemos de dos tipos: originales y

filtrados.

Los modelos de la carga están basados en cargas de trabajos de grandes sistemas paralelos los cuales se encuentran en estado activo. Cada modelo genera una carga con diferentes características. Los modelos que son utilizados en este trabajo son Feitelson96 [6] y Lublin99 [7], los cuales generan trabajos de tipo Rigid que especifican el número de procesadores que necesitan, así como el tiempo de ejecución. La carga sintética es un conjunto de trabajos con parámetros generados en forma aleatoria basada en algún tipo de distribución. Sin embargo, aun contando con este tipo de cargas para la simulación, no representan realmente a usuarios del C-Grid ya que como mencionamos anteriormente debe representar distintos usuarios, distintas cargas de trabajo, etc. por lo que pueden afectar la exactitud de la simulación y la evaluación de las estrategias de calendarización para el ambiente de C-Grid. Algunos problemas con la carga de trabajo para la evaluación de las estrategias en el C-Grid son: No existen cargas registradas para el grid computacional; No existen modelos de carga para el grid computacional; Variedad de parámetros de las tareas.

Como propuesta para solucionar este problema proponemos un modelo de carga para grid, haciendo una mezcla de logs reales con normalización de tiempos de release basado en el tiempo de inicio de cada log y zona horaria del sitio de este log.

3.1. Carga de trabajo para C-Grid

Sin embargo, aun contando con este tipo de cargas para la simulación, no representan realmente a usuarios del C-Grid ya que como mencionamos anteriormente debe representar distintos usuarios, distintas cargas de trabajo, etc. por lo que pueden afectar la exactitud de la simulación y la evaluación de las estrategias de calendarización para el ambiente de C-Grid.

Algunos problemas con la carga de trabajo para la evaluación de las estrategias en el C-Grid son:

- No existen cargas registradas para el grid computacional.
- No existen modelos de carga para el grid computacional
- Variedad de usuarios (variedad de parámetros de las tareas).

Como propuesta para solucionar este problema proponemos un modelo de carga para grid, haciendo una mezcla de logs reales con normalización de tiempos de release basado en el tiempo de inicio de cada log y zona horaria del sitio de este log.

IV. ANÁLISIS DE LOGS

Para simular la calendarización en un ambiente de cómputo paralelo típicamente utilizábamos tres tipos de cargas: Logs, Modelos, y carga sintética, sin embargo, estos tipos de cargas no representan realmente a usuarios del C-Grid. Por lo que pueden afectar la exactitud de la simulación y la evaluación de las estrategias de calendarización en el ambiente C-Grid. Una carga de trabajo para C-Grids tiene que estar constituida por tareas de distintos usuarios en distintos lugares del mundo y

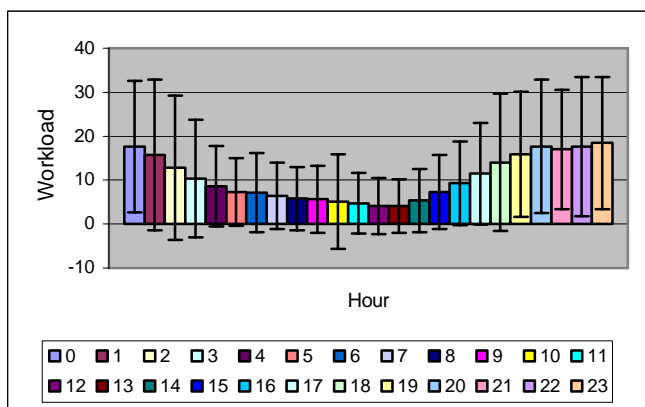


Figura 2a. Histograma del “l_sdsc_blue_cln” considerando el tiempo de inicio de tareas.

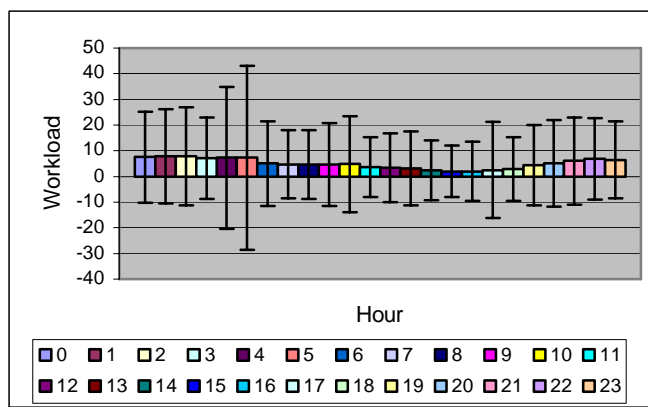


Figura 3a. Histograma del “l_osc_cln” considerando el tiempo de inicio de cada tarea.

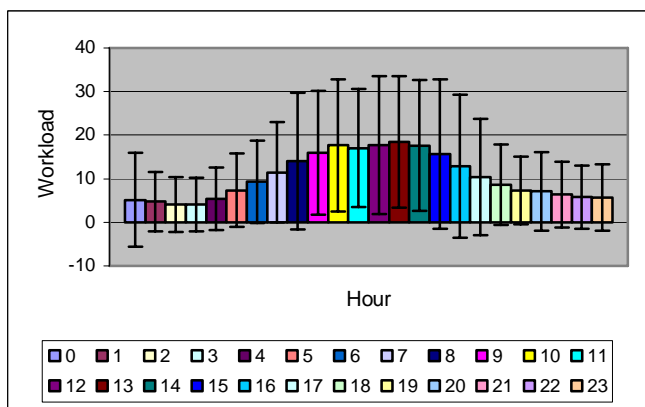


Figura 2b. Histograma del “l_sdsc_blue_cln” normalizada por tiempo de inicio + hora local + zona horaria (UTC).

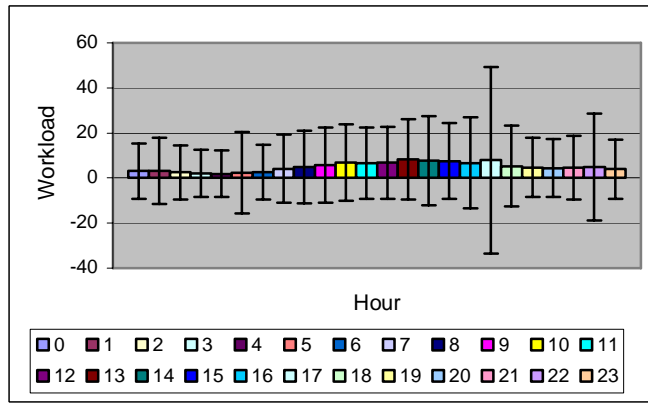


Figura 3b. Histograma del “l_osc_cln” normalizada por tiempo de inicio + hora local + zona horaria (UTC).

con distintos requerimientos de cómputo. En este capítulo analizaremos como cambia la distribución de tareas durante las 24 horas del día por un periodo de tiempo para distintos tipo de cargas de trabajo.

4.1. l_sdsc_blue_cln

Las Figuras 2a, 2b presentan la distribución promedio en 24 horas de carga de trabajo del log “l_sdsc_blue_cln”, perteneciente a SDSC Blue Horizon (Centro de Súpercomputo en San Diego), que contiene 250440 tareas registradas durante un periodo de tiempo de dos años con dos meses aproximadamente.

Como se muestra en la Figura 2a, el mayor numero de carga se presenta a partir de las 15 horas. Como se muestra en la Figura 2b, se puede ver que la carga de trabajo aumenta y mayor que la que se observa en la figura 2a, el numero de carga empieza a crecer a partir de las 5 horas.

4.2. l_osc_cln

Las Figuras 3a, 3b, presentan la distribución promedio en 24 horas de carga de trabajo del log “l_osc_cln”, perteneciente al centro de Súpercomputo de Ohio (OSC), que contiene 80714 tareas registradas durante un periodo de tiempo de un año con dos meses aproximadamente.

4.3. l_lanl_cm5_cln

Las Figuras 4a, 4b presentan la distribución promedio en 24 horas de carga de trabajo del log “l_lanl_cm5_cln”, perteneciente a los Alamos National Lab (LANL) que contiene 201385 tareas registradas durante un periodo de tiempo de un año con tres meses aproximadamente.

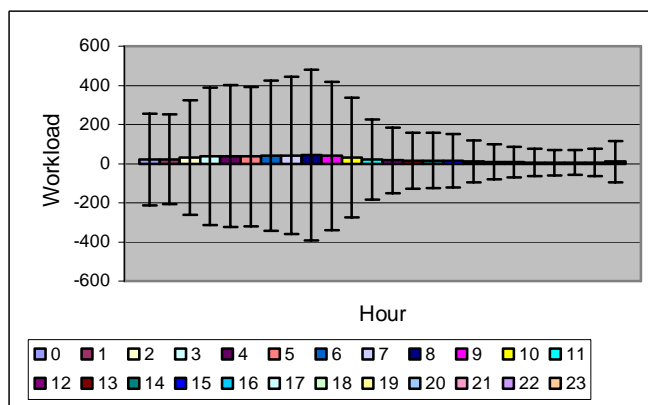


Figura 4a. Histograma del “l_lanl_cm5_cln” considerando el tiempo de inicio de cada tarea.

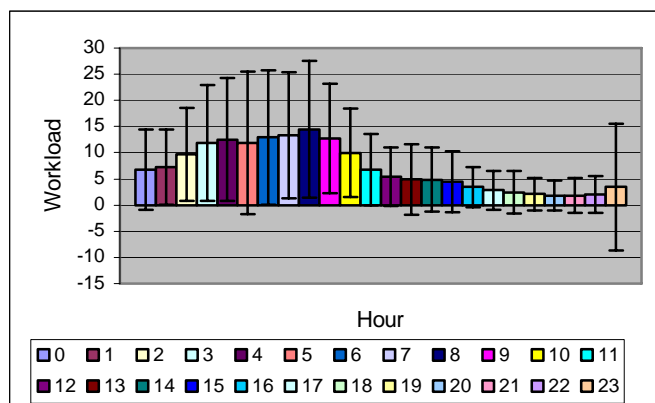


Figura 4b. Histograma del “l_lanl_cm5_cln” normalizada por tiempo de inicio (hora local) y zona horaria (UTC).

V. CARGA DE TRABAJO PARA UN GRID COMPUTACIONAL

Para crear una carga de trabajo para un grid computacional proponemos mezclar logs que pertenecen a diferentes centros de cómputo participantes en un C-Grid. Por ejemplo, creamos una carga con tres recursos pertenecientes a tres centros de cómputo mencionados anteriormente, los cuales analizamos en capítulos 3.1-3.3 (Véase las Figuras 5a, y 5b), presentamos dos maneras de generar la mezcla: 1. Considerando el tiempo de inicio de cada tarea (Figura 5a). 2. Normalizando por tiempo de inicio (local) y zona horaria (UTC) (Fig. 5b).

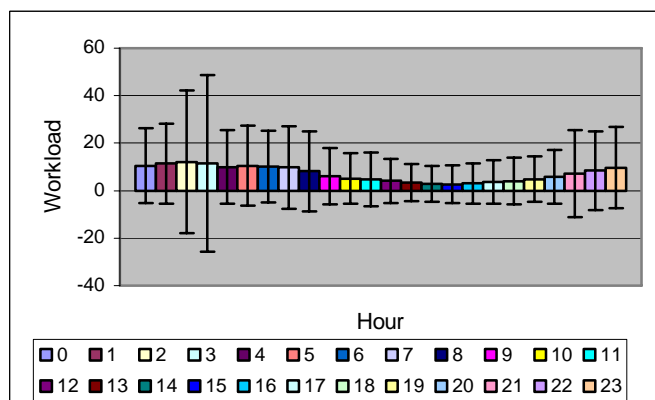


Figura 5a. Histograma de carga de trabajo normalizada por tiempo de inicio (hora local)

VI. ANÁLISIS EXPERIMENTAL DE LAS ESTRATEGIAS DE CALENDARIZACIÓN CON LOG GENERADO

Para analizar como influye el tipo de carga de trabajo en un simulador de ambiente C-Grid, haremos tres experimentos utilizando cargas de trabajo con características diferentes. Todos los experimentos se llevarán a cabo con las siguientes combinaciones de selección de trabajos, selección de recursos y calendarización local: FIFO-MCT-FCFS-FCFS, FIFO-MLB-FCFS, FIFO-MLP-FCFS, FIFO-MPL-FCFS, FIFO-MWT-FCFS-FCFS, FIFO-Random-FCFS. Los experimentos serán realizados con los siguientes parámetros de entrada: 50 experimentos, 1000 trabajos. Usaremos un C-Grid virtual con 7 recursos que tienen datos de supercomputadoras reales (Tabla 2):

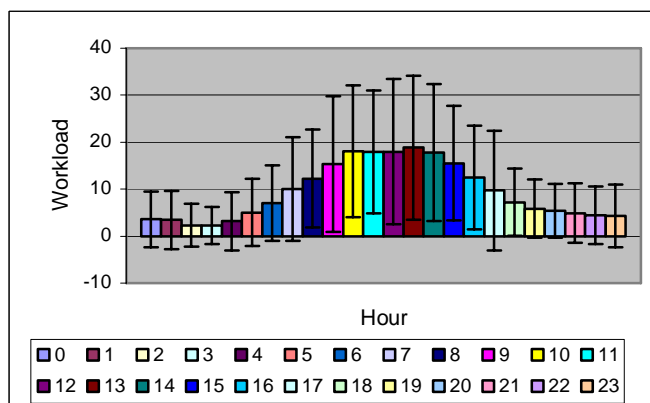


Figura 5b. Histograma de carga de trabajo normalizada por tiempo de inicio (hora local) y zona horaria UTC

Realizamos la simulación del ambiente C-Grid y evaluamos las diferentes combinaciones de estrategias de calendarización tomamos como cargas de trabajo la mezcla de logs y mezclamos aleatoriamente: el modelo FEITELSON96; la mezcla de 3 logs (SDSC_BLUE_HORIZON, OSC_CLUSTER, LANL_CM_5) normalizados por tiempo de inicio y zona horaria (UTC).

	Recurso	Instalación	Procs
1	IBM SP2	San Diego Supercomputer Center (SDSC)	128
2	Intel iPSC/860	NASA Ames Research Center	128
3	Linux Cluster	Ohio Supercomputer Center (OSC)	178
4	Intel Paragon	San Diego Supercomputer Center (SDSC)	352 (416 total)
5	IBM SP2	Cornell Theory Center (CTC)	430 (512 total)
6	CM-5	Los Alamos National Lab (LANL)	1024
7	IBM SP3	Blue Horizon, San Diego Supercomputer Center (SDSC)	1152

Tabla 2. Características del Grid

6.1 Resultados experimentales de evaluación basados en el criterio de evaluación orientado a algoritmo de calendarización

Las Figuras 6a, 6b, 6c representan los resultados de evaluación basados en el criterio de evaluación orientado a algoritmo de calendarización. Como se muestra la combinación de estrategias que presentó un mejor desempeño es FIFO-Min_CT-FCFS-FCFS, y la estrategia que presentó mal desempeño es FIFO-Random-FCFS en todos los experimentos.

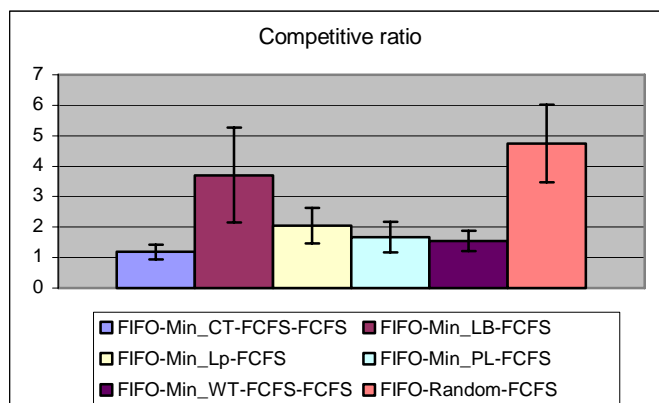


Figura 6a. Coeficiente de desempeño con carga de trabajos sintética

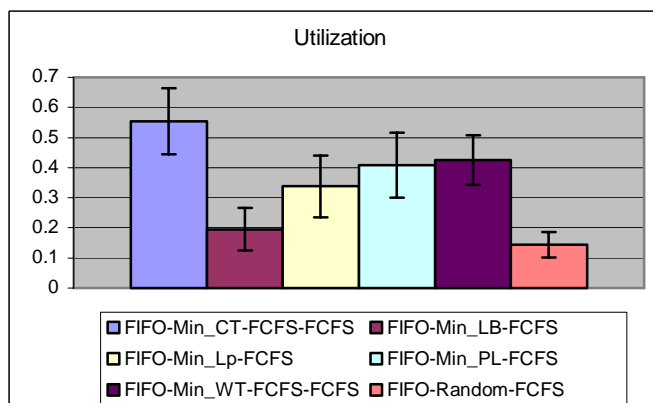


Figura 7a. Utilización con carga de trabajos sintética

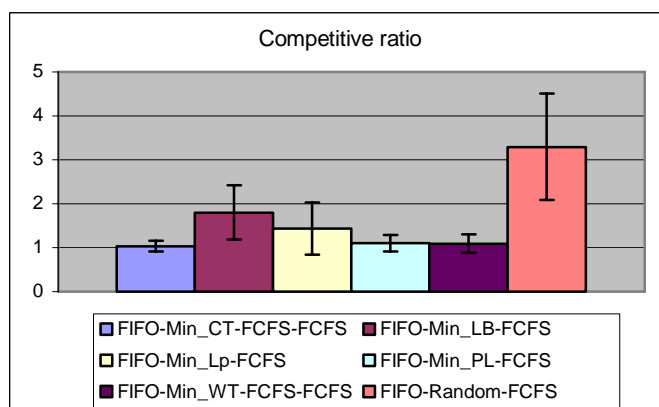


Figura 6b. Coeficiente de desempeño con carga de trabajos FEITELSON96

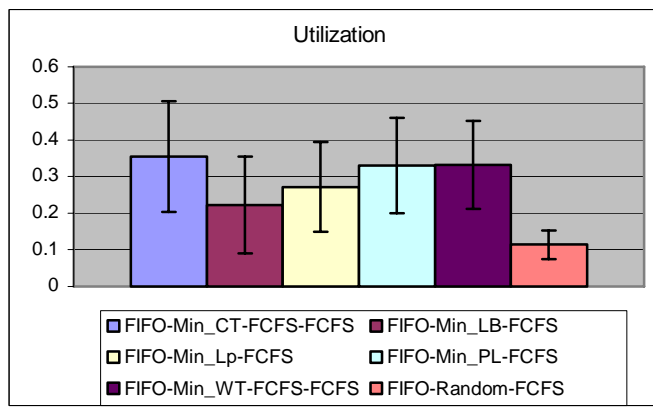


Figura 7b. Utilización con carga de trabajos basada en un modelo de FEITELSON96

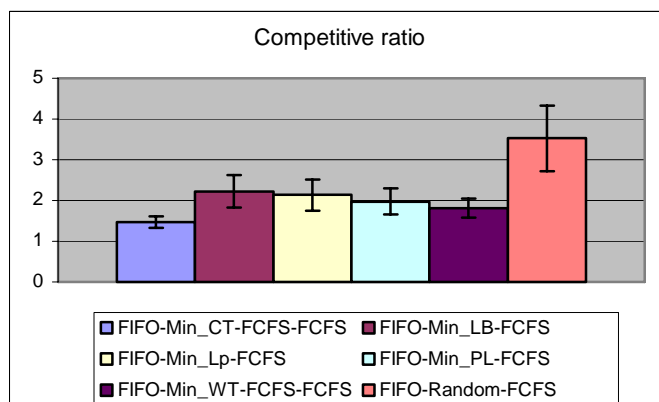


Figura 6c. Coeficiente de desempeño con carga de trabajos basada en la mezcla de 3 logs

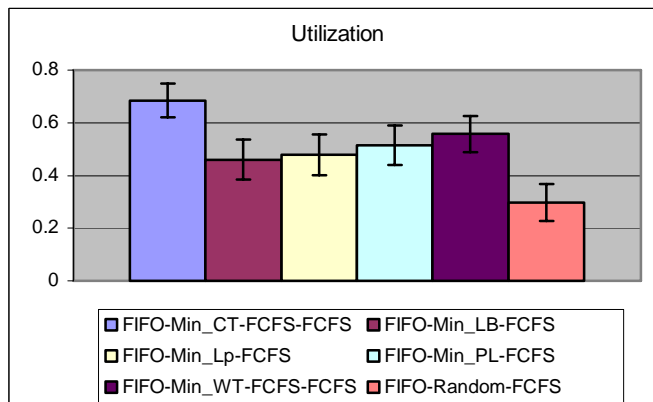


Figura 7c. Utilización con carga de trabajos basada en la mezcla de 3 logs

6.2 Resultados experimentales de evaluación basados en los criterios de evaluación orientados a sistema

Las Figuras 7a, 7b, 7c, representan resultados de simulación con el criterio de utilización. Las Figuras muestran que la combinación de estrategias que presento una mayor utilización del sistema es FIFO-Min_CT-FCFS-FCFS, contrario a este resultado es la estrategia FIFO-Random-FCFS en todos experimentos

Las Figuras 8a, 8b, 8c representan resultados de simulación con el criterio de rendimiento del procesamiento. Las Figuras muestran que la combinación de estrategias que presentó el mejor rendimiento del procesamiento del Grid es FIFO-Min_CT-FCFS-FCFS, y la que presentó un menor rendimiento es FIFO-Random-FCFS en todos los experimentos.

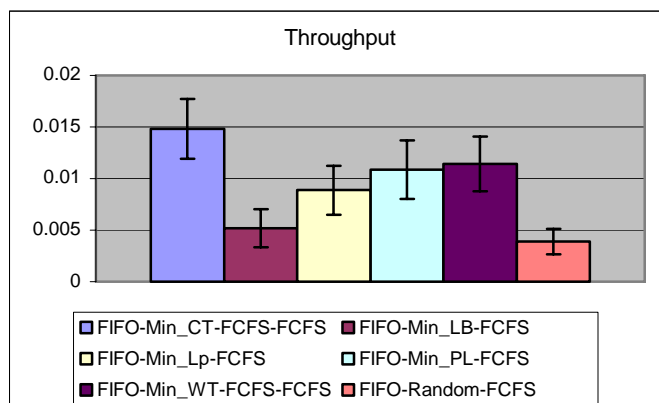


Figura 8a. Rendimiento del procesamiento con carga de trabajos sintética

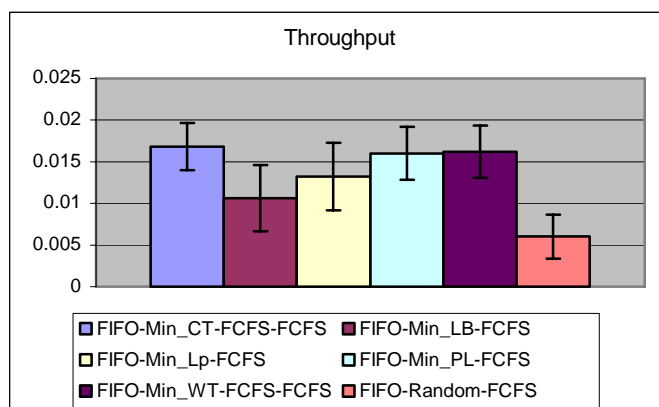


Figura 8b. Rendimiento del procesamiento con carga FEITELSON96

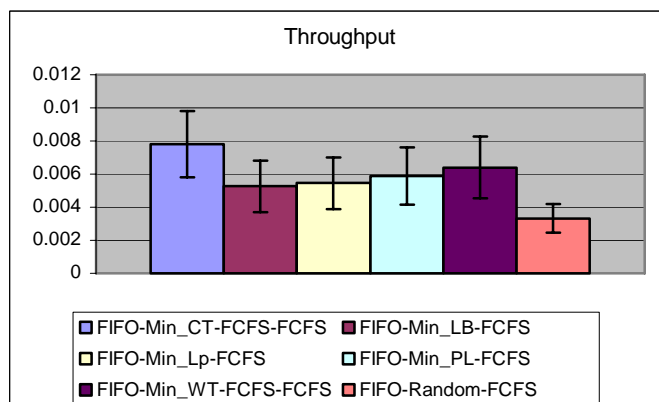


Figura 8c. Rendimiento del procesamiento con carga la mezcla de 3 logs

6.3 Resultados experimentales de evaluación basados en los criterios de evaluación orientados a usuario

Las Figuras 9a, 9b y 9c representan resultados de simulación con el criterio de tiempo de respuesta. Las figuras muestran que las combinaciones de estrategias que tienen los menores tiempos de respuesta promedio son FIFO-Min_WT-FCFS-FCFS y FIFO-Min_PL-FCFS, y la que presentan

resultados desfavorables son FIFO-Min_LB-FCFS y FIFO-Random-FCFS.

Las Figuras 10a, 10b y 10c representan resultados de simulación con el criterio de tiempo de espera. Las Figuras muestran que la combinación de estrategias que presentan el menor tiempo de respuesta promedio son FIFO-Min_WT-FCFS-FCFS, FIFO-Min_PL-FCFS, y la que presentó un mayor tiempo de espera son FIFO-Min_LB-FCFS, FIFO-Random-FCFS.

Las Figuras 11a, 11b, 11c representan resultados de simulación con el criterio de coeficiente de respuesta. Las Figuras muestran que la combinación de estrategias que presentó el menor coeficiente de respuesta son FIFO-Min_WT-FCFS-FCFS y FIFO-Min_PL-FCFS, y la que presentó un mayor coeficiente de respuesta son FIFO-Min_LB-FCFS y FIFO-Random-FCFS.

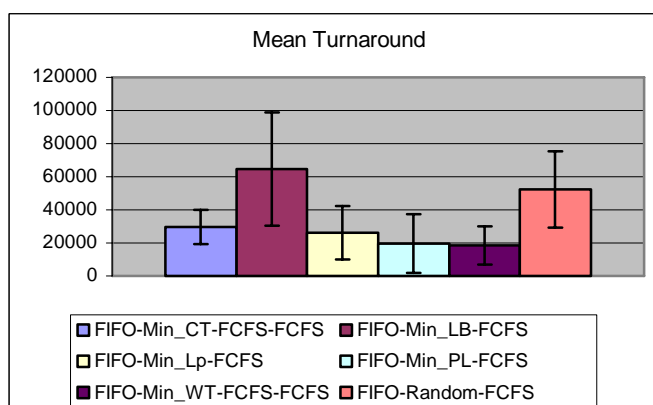


Figura 9a. Tiempo de respuesta promedio con carga de trabajos sintética

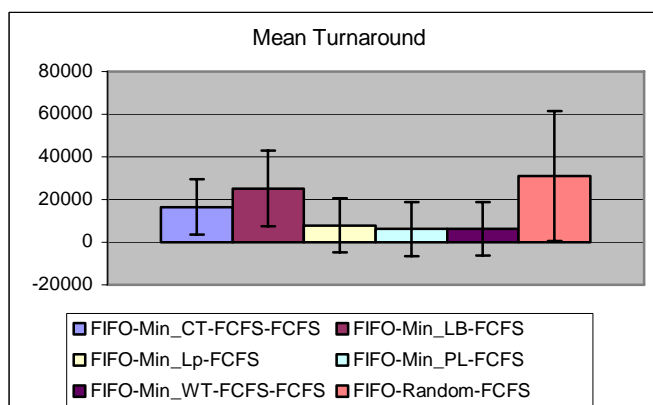


Figura 9b. Tiempo de respuesta promedio con carga FEITELSON96

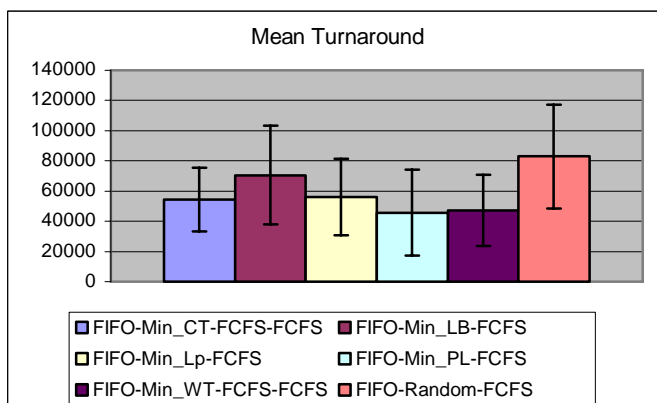


Figura 9c. Tiempo de respuesta promedio con carga la mezcla de 3 logs

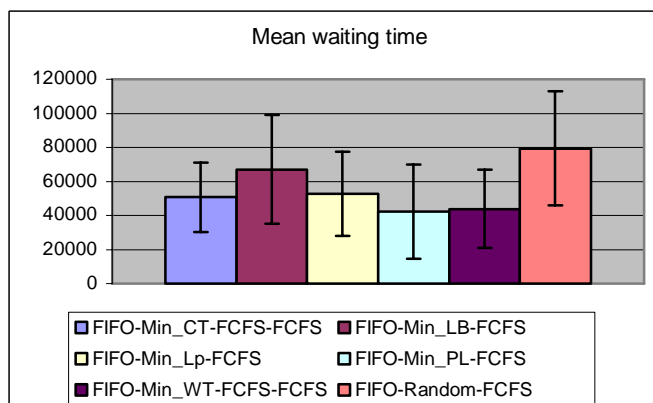


Figura 10c. Tiempo de espera con carga la mezcla de 3 logs

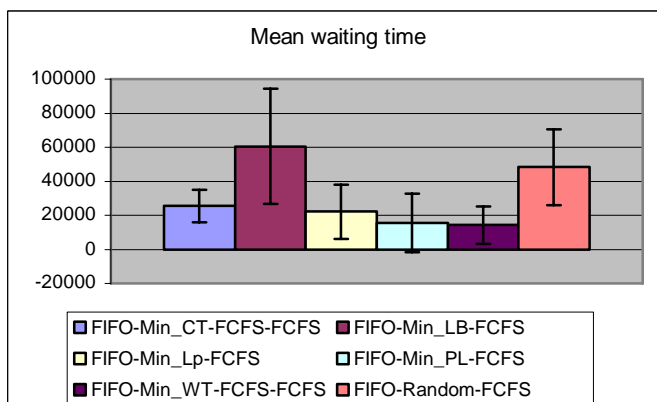


Figura 10a. Tiempo de espera con carga de trabajos sintética

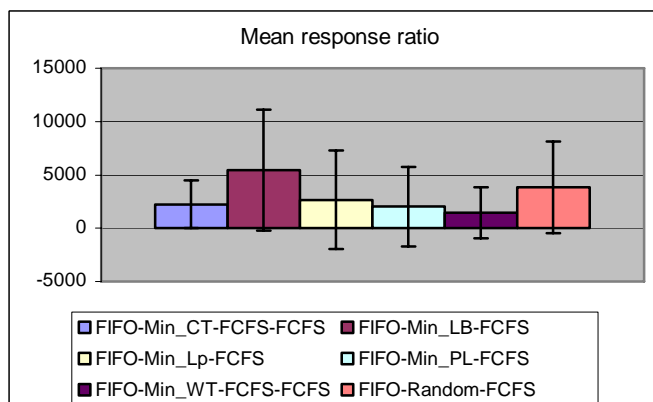


Figura 11a. Coeficiente de respuesta con carga de trabajos sintética

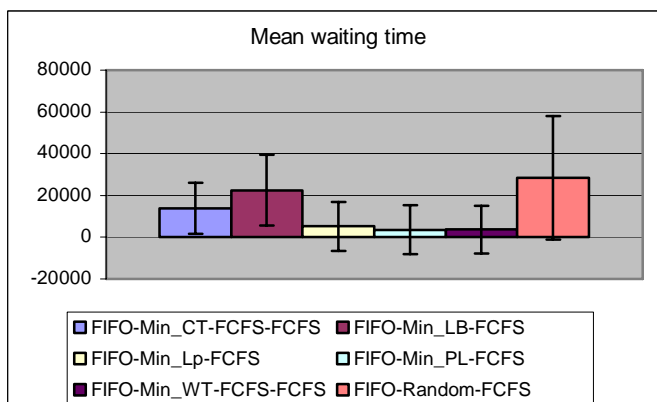


Figura 10b. Tiempo de espera con carga FEITELSON96

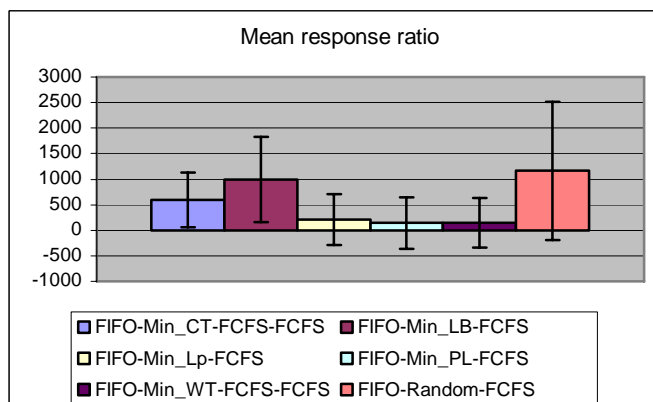


Figura 11b. Coeficiente de respuesta con carga FEITELSON96

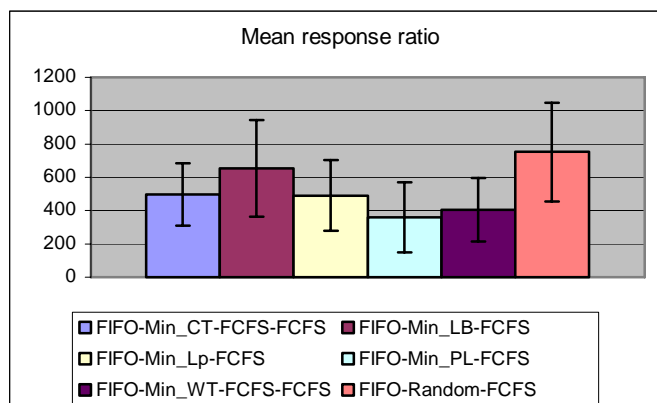


Figura 11c. Coeficiente de respuesta con carga la mezcla de 3 logs

VII. CONCLUSIÓN

Para la evaluación de la calendarización en un ambiente Grid, utilizamos distintos tipos de carga: carga sintética, modelo Feitelson96, y carga generada para C-Grid. Evaluamos las combinaciones de las estrategias de calendarización basándonos en 3 grupos de criterios de evaluación.

Desde el punto de vista del algoritmo de calendarización, como es el coeficiente de desempeño (competitive ratio)

Desde el punto de vista del sistema, como son la utilización (utilization) y el rendimiento del procesamiento (throughput)

Desde el punto de vista del usuario, como son el tiempo de respuesta promedio (mean turnaround), tiempo de espera (mean waiting time), y coeficiente de respuesta (mean response ratio).

La Tabla 3 muestra las estrategias de calendarización que presentaron mejores resultados para ciertos criterios de evaluación dependiendo del tipo de carga utilizada en el experimento. Las columnas representan los criterios de evaluación, y las filas la combinación de estrategias de calendarización. La letra indica el tipo de carga de trabajo que se uso en la simulación.

La Tabla 4 muestra las estrategias de calendarización que presentaron los peores resultados para ciertos criterios de evaluación dependiendo del tipo de carga utilizada en el experimento.

Estrategias	1	2	3	4	5	6
FIFO-Random-FCFS						
FIFO-Min_PL-FCFS			a, b, c	c	b, c	b, c
FIFO-Min_WT-FCFS-FCFS	a, b, c			a, b	a	a
FIFO-Min_CT-FCFS-FCFS		a, b, c				

Tabla 3. Estrategias con mejores resultados

Estrategias	1	2	3	4	5	6
FIFO-Random-FCFS	a, b, c	a, b, c	a, b, c	b, c	b, c	b, c
FIFO-Min_LB-FCFS				a	a	a
FIFO-Min_CT-FCFS-FCFS						

Tabla 4. Estrategias con peores resultados

Se analizaron las cargas de trabajo para computadoras paralelas y la carga generada para grid, se observo que una carga generada para grid contiene una mejor distribución de las tareas a lo largo del día por lo que se aproxima mas a la realidad en cuanto al tipo de carga de trabajo que llega a un Grid Computacional, al evaluar las estrategias de Calendarización con la carga para computadoras paralelas, y clusters, como son logs de sitios específicos y modelos de carga, y carga generada para grid, se observo que los modelos de carga de trabajo y la carga de trabajo generada para grid tenían mas similitudes en los resultados obtenidos, contrario a los resultados que se obtuvieron al evaluar las estrategias de Calendarización utilizando carga sintética, por el momento concluimos que el tipo de carga que se utiliza interviene en los resultados de la evaluación de las estrategias de Calendarización en el grid computacional.

REFERENCIAS

- [1]Rajkumar Buyya, David Abramson, and Jonathan Giddy. *An Economy Driven Resource Management Architecture for Global Computational Power Grids*. www.csse.monash.edu.au/~davida/papers/GridEconomy.pdf
- [2]Aida K, Takefusa A, Nakada H, Matsuoka S, Sekiguchi S, Nagashima U. Performance evaluation model for scheduling in a global computing system. *The Int. Journal of High Performance Comp. Applications* 2000; **14**(3).
- [3]Song H, Liu X, Jakobsen D, Bhagwan R, Zhang X, Taura K, Chien A. The MicroGrid: A scientific tool for modelling computational Grids. *Proceedings of IEEE Supercomputing (SC 2000)*, Dallas, TX, 4–10 November 2000.
- [4]Casanova H. Simgrid: A toolkit for the simulation of application scheduling. *Proc.1st IEEE/ACM Int.Symposium on Cluster Computing and the Grid*, Australia, IEEE 2001.
- [5]Rajkumar Buyya and Manzur Murshed, GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, *The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, V.14, 13-15, Wiley Press, 2002.
- [6]D. G. Feitelson, Packing Schemes for Gang Scheduling. In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson, L. Rudolph (Eds.), Springer-Verlag 1996, LNCS. vol. 1162, pp. 89-110.
- [7]U. Lublin, D. G. Feitelson, The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs, Tec. Report 2001-12, School of Computer Science and Engineering, The Hebrew University of Jerusalem, Oct 2001.



Elisa Salazar nació en la Ciudad de México el 12 de abril de 1981, es pasante de la Licenciatura en Ciencias de la Computación en la Universidad de Sonora. Ha asistido a dos veranos de investigación científica por parte del programa Delfin, en ambas ocasiones trabajo con el Dr. Andrei Tchernykh en el Centro de Investigación Científica y de Educación Superior de Ensenada. Actualmente esta realizando su tesis de licenciatura con el Dr. Andrei Tchernykh y con la M.C. Edelmira Rodríguez. Las áreas de estudio que más le interesan actualmente son el cómputo paralelo, grids y sistemas distribuidos de tiempo real.



Dr. Andrei Tchernykh is a researcher in the Computer Science Department, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Baja California, México since 1995. He graduated from the Sevastopol Technical University (*with honor*) in 1975. He received his Ph.D. degree in Computer Science from the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences (Moscow, Russia), in 1986. His main interests include parallel and cluster computing, on-line scheduling, real time systems, and resource

optimization in GRID computing.